

CLAIMS

What is claimed is:

1 1. A method for enabling a front-end load balancing function to a cluster of servers
2 functioning as an Internet site for serving end-users, said front-end load balancing
3 function and said end-users establishing transmission control protocol (TCP)
4 connections, said method comprising the steps of:

 spreading said front-end load balancing function over more than one individual
 load balancer (ILB);

 enabling each ILB to consistently self-assert, for said front-end load balancing
 function, an ILB owner for each one of said TCP connections;

 processing an ILB-owned TCP connection on behalf of said load balancing
 function in said each ILB, for each one of said TCP connections owned by each said ILB;
 and

 handing off an unassigned TCP connection to said ILB owner in each said ILB,
 for each one of said TCP connections not owned by each said ILB.

1 2. The method according to claim 1, wherein said enabling step further includes
2 steps of:

 consistently self-asserting, for said front-end load balancing function, a back up
 ILB owner for each one of said TCP connections; and

5 utilizing said back up ILB owner to process said ILB owned TCP connection
6 should said ILB owner become incapable of processing said ILB owned TCP connection.

1 **3.** The method according to claim 2, wherein said enabling step, further includes
2 steps of:

3 using a connection unique identifier for each one of said TCP connections;

4 using an ILB unique identifier for each said ILB;

5 computing a score for each said ILB;

6 obtaining a set of scores;

7 ranking said set of scores, said ranking step further includes:

8 designating said ILB having a best ranked score as said ILB owner;

9 designating said ILB having a second best ranked score as said back up ILB
10 owner.

1 **4.** The method according to claim 3, wherein said method further includes a step of:

2 adding a connection unique identifier use step wherein said connection unique
3 identifier is a twelve byte quadruplet including a TCP destination port, an IP destination
4 address, a TCP source port, an IP source address.

1 **5.** The method according to claim **4**, wherein said method further includes a step of:

2 using said ILB unique identifier wherein said ILB unique identifier is an IP
3 address of said ILB.

6 **6.** The method according to claim **4**, wherein said method further includes a step of:

 using said ILB unique identifier wherein said ILB unique identifier is an index
 which is unique within said front-end load balancing function.

7 **7.** The method according to claim **3**, wherein said ranking step further includes a
 step of:

3 ranking on the basis of the arithmetical values of said set of scores.

1 **8.** The method according to claim **3**, wherein said designating steps further includes
2 steps of:

3 selecting the largest arithmetical value of said set of scores as said best ranked
4 score; and

5 selecting the second largest arithmetical value of said set of scores as said second
6 best ranked score.

1 **9.** The method according to claim **3**, wherein said step of computing a score further
2 includes a step of:

3 calculating a cyclic redundancy check (CRC) code over said ILB unique identifier
4 concatenated to said connection unique identifier.

1 **10.** The method according to claim **1**, wherein said method further includes a step
2 of:

3 designating a particular server, out of said cluster of servers due to process each
4 said ILB owned TCP connection, with a lookup table of owned connections (LTOC)
5 included in each said ILB.

1 **11.** The method according to claim **10** wherein said method further includes a step
2 of:

3 including in each said LTOC, for each ILB owned TCP connection, said backup
4 ILB owner.

1 **12.** The method according claim **1**, wherein said method further includes a step of:

2 including a cache of all least recently used associations formed, within said front-
3 end load balancing function, between each one of said TCP connections with an
4 individual server out of said cluster of servers, in each of said ILB.

1 **13.** The method according to claim **3**, upon receiving a TCP synchronous idle
2 character (SYN) Packet in a receiving ILB, wherein said method further includes steps

3 of:

4 computing said set of scores;

5 determining said ILB owner and said back up ILB owner;

6 checking if said receiving ILB is said ILB owner;

7 selecting an individual server to process a new TCP connection, in response to
8 determining that said receiving ILB is said ILB owner;

9 forwarding said TCP SYN packet to said individual server;

10 broadcasting a control packet within said front-end load balancing function
11 informing of a new formed association between said new TCP connection, said
12 individual server and said back up ILB owner in all ILBs receiving said broadcast control
13 packet;

14 optionally caching said new formed association;

15 testing if said broadcast receiving ILB is selected said backup ILB owner;

16 moving forward directly to said completing step, in response to determining that
17 said broadcast receiving ILB is not selected said backup ILB owner;

18 storing in said LTOC of said broadcast receiving ILB that, for said new TCP
19 connection, said broadcast receiving ILB is said back up ILB owner, in response to
20 determining that said broadcast receiving ILB is selected said backup ILB owner;

forwarding said TCP SYN packet to said ILB owner, in response to determining that said receiving ILB is not said ILB owner;

forwarding said TCP SYN packet to said individual server;

broadcasting a control packet within said front-end load balancing function informing of a new formed association between said new TCP connection, said individual server and said back up ILB owner;

in all ILBs receiving said broadcast control packet;

optionally caching said new formed association;

testing if said broadcast receiving ILB is selected said backup ILB owner;

moving forward directly to said completing step, in response to determining that said broadcast receiving ILB is not selected said backup ILB owner; and

storing in said LTOC of said broadcast receiving ILB that, for said new TCP connection, said broadcast receiving ILB is said back up ILB owner, in response to determining that said broadcast receiving ILB is selected said backup ILB owner.

14. The method according to claim 3, upon receiving a transmission control protocol (TCP) Packet other than a synchronous idle character (SYN), in said receiving individual load balancer (ILB), said method further comprising the steps of:

computing said set of scores;

determining said ILB owner and said back up ILB owner;

6 checking if said receiving ILB is said ILB owner or said back up ILB owner of
7 a previously established TCP connection;

8 retrieving in said LTOC of said receiving ILB a corresponding entry for said
9 previously established TCP connection, in response to said receiving ILB is said ILB
10 owner or said backup ILB owner of a previously established TCP connection;

11 forwarding said TCP packet to said individual server according to said LTOC
12 corresponding entry;

13 testing if said TCP packet is a FIN or RST packet;

14 moving forward directly to said completing step, in response to determining that
15 said TCP packet is not a FIN or RST packet;

16 broadcasting an end of connection control packet within said front-end load
17 balancing function informing all ILBs that said previously established TCP connection
18 has terminated thus, that each said cache and said LTOC of said back up ILB owner
19 and/or said ILB owner must be updated accordingly, in response to determining that said
20 TCP packet is a FIN or RST packet;

21 looking up said cache of said receiving ILB for an entry corresponding to said
22 previously established TCP connection, in response to determining that said receiving
23 ILB is not said ILB owner nor said back up ILB owner of a previously established TCP
24 connection;

25 forwarding said TCP packet to said individual server according to said LTOC
26 corresponding entry, in response to finding an entry corresponding to said previously
27 established TCP connection;

28 testing if said TCP packet is a FIN or RST packet;

29 moving forward directly to said completing step, in response to determining that

30 said TCP packet is not a FIN or RST packet;

31 broadcasting an end of connection control packet within said front-end load
32 balancing function informing all ILBs that said previously established TCP connection
33 has terminated thus, that each said cache and said LTOC of said back up ILB owner
34 and/or said ILB owner must be updated accordingly, in response to determining that said
35 TCP packet is a FIN or RST packet; and

36 forwarding said TCP packet to said ILB owner, in response to not finding an entry
37 corresponding to said previously established TCP connection.

15. The method according to claim 14, wherein said method further comprising the
step of:

forwarding to said ILB owner said received packet other than a SYN when
receiving ILB has no cache, in response to determining that said receiving ILB is not said
ILB owner nor said backup ILB owner of a previously established TCP connection.

1 16. The method according to claim 1, wherein said method further comprising the
2 step of:

3 broadcasting ID messages regularly from each said ILB in order to keep all other
4 said ILBs aware of their respective status while each said ILB actively participating, at
5 any given instant, in said front-end load balancing function.

1 17. The method according to claim 16, wherein upon listening for the reception of
2 said ID messages, in said receiving ILB said method further comprising the steps of:

checking if a new ILB has joined said front-end load balancing function, in response to receiving a new ID message;

continuing ID message monitoring, in response to determining that said new ILB has not joined said front-end load balancing function;

re-computing scores including said new ILB, in response to determining that said new ILB has joined said front-end load balancing function for each one of said TCP connections currently handled by said receiving ILB as said ILB owner or as said backup owner;

updating a transfer table, said updating step said transfer table further comprising:

adding said ILB owned TCP connection in said transfer table as now owned by said new ILB, in response to determining if said new ILB is elected to become said ILB owner and said receiving ILB is elected to become said back up ILB owner;

changing, in said LTOC of said receiving ILB, state of current TCP connection;

adding said ILB owned TCP connection in said transfer table as now backup owned by said new ILB, in response to determining if said receiving ILB remains to be said ILB owner and said new ILB is elected to become said back up ILB owner;

adding said ILB owned TCP connection in said transfer table as now back up owned by said new ILB, in response to determining said new ILB is elected to become said back up ILB owner and said receiving ILB is no longer said ILB owner or said back up ILB owner;

deleting, in said LTOC of said receiving ILB, current TCP connection; and

transferring said transfer table to said new ILB.

18. The method according to claim 17, wherein upon listening for the reception of

2 said ID messages, in said receiving ILB, said method further comprising the steps of:

3 checking if a former ILB has left said front-end load balancing function, in
4 response to receiving a new ID message;

5
6 continuing ID message monitoring, in response to determining if a former ILB
7 has not left said front-end load balancing function;

8 flushing said cache of said receiving ILB of all entries corresponding to said
9 former ILB, in response to determining if a former ILB has left said front-end load
10 balancing function;

11 updating said transfer table, for each one of said TCP connections currently
12 handled by said receiving ILB as said ILB owner or as said back up ILB owner, said step
13 of updating said transfer table further comprising the step of:

14 re-computing a new back up ILB owner among remaining ILBs, in response to
15 determining said receiving ILB is said ILB owner and said former ILB was said backup
16 ILB owner;

17 adding in said transfer table said new back up ILB owner;

18 changing, in said LTOC of said receiving ILB, state of current TCP connection
19 so as said receiving ILB becomes a new ILB owner, in response to determining said
20 receiving ILB is said back up ILB owner and said former ILB was said ILB owner;

21 re-computing a new back up ILB owner;

22 adding in said transfer table said new ILB owner and said new back up ILB
23 owner;

24 transferring said transfer table to all remaining ILBs.

1 **19.** A system for enabling a front-end load balancing function to a cluster of servers
2 together implementing an Internet site for serving end-users, said front-end load
3 balancing function and said end-users establishing TCP connections, said system
4 comprising:

5 means for spreading said front-end load balancing function over more than one
6 individual load balancer (ILB);

7 means for enabling each ILB to consistently self-assert, for said front-end load
8 balancing function, an ILB owner for each one of said TCP connections;

9 means for processing an ILB-owned TCP connection in behalf of said load
10 balancing function in said each ILB, for each one of said TCP connections owned by said
11 each ILB; and

12 means for handing off an ILB not owned TCP connection to said ILB owner in
13 each said ILB, for each one of said TCP connections not owned by said each ILB.

1 **20.** A computer-readable medium comprising instructions for enabling a front-end
2 load balancing function to a cluster of servers together implementing an Internet site for
3 serving end-users, said front-end load balancing function and said end-users establishing
4 TCP connections, said instruction means comprising:

5 instruction means for spreading said front-end load balancing function over more
6 than one individual load balancer (ILB);

7 instruction means for enabling each ILB to consistently self-assert, for said front-

8

9

12

13

14

1999

15

1100

Abstract

22

Abstract

0.0000

iii

5

10

55

Strong

50

2007